

Tilburg University

Job Scheduling, Cooperation and Control

Calleja, P.; Estevez Fernandez, M.A.; Borm, P.E.M.; Hamers, H.J.M.

Publication date:
2004

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Calleja, P., Estevez Fernandez, M. A., Borm, P. E. M., & Hamers, H. J. M. (2004). *Job Scheduling, Cooperation and Control*. (CentER Discussion Paper; Vol. 2004-65). Operations research.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



No. 2004–65

JOB SCHEDULING, COOPERATION, AND CONTROL

By P. Calleja, M.A. Estévez-Fernández, P.E.M. Borm,
H.J.M. Hamers

July 2004

ISSN 0924-7815

Job scheduling, cooperation, and control

Pedro Calleja^{1,2}, Arantza Estévez-Fernández^{3,4}, Peter Borm³ and Herbert Hamers³

Abstract:

This paper considers one machine job scheduling situations or sequencing problems, where clients can have more than a single job to be processed in order to get a final output. Moreover, a job can be of interest for different players. This means that one of the main assumptions in classic sequencing problems is dropped: the one to one correspondence between clients and jobs. It is shown that the corresponding cooperative games are balanced for specific types of cost criteria.

Keywords: cooperative game theory, scheduling, balancedness.

JEL classification: C71

1 Introduction

In a job scheduling situation or sequencing problem a number of jobs has to be processed in some order on one or more machines in such a way that a specific cost criterion is minimized. Job scheduling situations can be classified on the basis of many features. We mention the number of machines, the specific properties of the machines (e.g., parallel, serial), the chosen cost criterion (e.g., maximum completion time, weighted completion time), restrictions on the jobs (e.g., ready times, due dates) and possibly the specific order in which the jobs have to be processed on the machines (e.g., job-shop, flow-job).

By associating jobs to clients, a sequencing problem gives rise to a multi-active decision making problem. Each client incurs costs, depending on the completion times of his jobs. By assuming an initial order on the jobs, the first problem the clients jointly face is that of finding an optimal reordering of all jobs, i.e., a schedule maximizing joint cost savings. The subsequent problem is how to reallocate these cost savings in a fair way. This “fairness” problem can be analyzed from a game-theoretic point of view. By defining the value of a coalition of clients as the maximum costs it can save by means of an optimal admissible reordering, we obtain a cooperative sequencing game related to the sequencing problem. The core of this game provides insight in the allocation problem at hand since core elements provide a stable reallocation of the joint cost savings. A game is said to be balanced if it has a non-empty core.

The above game-theoretic approach to sequencing situations was initiated by Curiel, Pederzoli and Tijs (1989) by considering the class of one-machine sequencing situations. The weighted completion cost criterion is used and it was shown that the corresponding sequencing games are convex, and thus balanced. In Curiel, Potters, Prasad, Tijs and Veltman (1994) one-machine sequencing situations are considered in which each agent has a weakly increasing cost function. They show that also in this extended setting the corresponding sequencing games are balanced.

¹Department of Economic, Financial, and Actuarial Mathematics, University of Barcelona, Av. Diagonal 690, 08034 Barcelona, Spain.

²Financial support has been given by the Generalitat de Catalunya through a BE grant from AGAUR

³CentER and Department of Econometrics and Operations Research, Tilburg University, P.O.Box 90153, 5000 LE Tilburg, The Netherlands

⁴Corresponding author. *E-mail address:* A.E.Fernandez@uvt.nl

Hamers, Borm and Tijs (1995) extend the class considered by Curiel et al. (1989) by imposing ready times on the jobs. The corresponding sequencing games are balanced and convex under some additional assumption. Similar results are obtained in Borm, Fiestras-Janeiro, Hamers, Sánchez and Voorneveld (2002) in which due dates are imposed on the jobs. Instead of imposing further restrictions on the jobs, van den Nouweland, Krabbenborg and Potters (1990), Hamers, Klijn and Suijs (1999) and Calleja, Borm, Hamers, Klijn and Slikker (2002) extend the number of machines. In each case balancedness was established for the corresponding games for specific instances. Finally, van Velzen and Hamers (2003) and Slikker (2003) consider relaxations in the notion of admissibility of reorderings in one machine sequencing situations and focus on balancedness. A recent review on sequencing games can be found in Curiel, Hamers and Klijn (2002).

Throughout the above mentioned literature it is assumed that there is a one to one correspondence between clients and jobs to be processed on a machine. Now this main assumption in sequencing models is dropped. This paper considers one machine sequencing situations where clients can have more than a single job to be processed. Moreover, a single job can be of interest to more than a single player. Think for instance in tasks that have to be processed by the CPU of a computer. One particular task could be of interest to different clients, and a particular client could be waiting for different tasks in order to obtain a final result. Another example can be found in a court where several legal proceedings have to be pronounced by an examining magistrate. Different people can be involved in the same case and a person can be involved in more than one case. A job is controlled by all players that are involved in it meaning that a coalition may change the position of a job only if all the players that are involved belong to the coalition. We focus on balancedness of the corresponding games and show that it is achieved when cost functions of players are additive with respect to the initial order on the jobs. A formal description of the model is presented in section 2. Section 3 is devoted to the balancedness of this kind of games.

2 The model

We consider one-machine sequencing problems, where there is a queue of jobs in front of a machine waiting to be processed. A job can be of interest to more than one player and every player can be involved in more than one job. The finite set of players is denoted by N . The finite set of jobs is denoted by M . The jobs in which the players are involved are described by a correspondence $J : N \twoheadrightarrow M$, where $J(i)$ denotes the set of jobs in which player i is involved. We assume $|J(i)| \geq 1$, i.e., each player is involved in at least one job. The set of players that are involved in job j is denoted by $N(j)$. We assume $|N(j)| \geq 1$, i.e., at least one player is involved in a job. This type of situations incorporates the classical one-machine sequencing situations, introduced by Curiel et al. (1989), where $|N| = |M|$ and J corresponds to a bijective function. Positions of the jobs in a queue (a processing order) are described by a bijection $\sigma : M \rightarrow \{1, \dots, |M|\}$, where $\sigma(j) = k$ means that job j is at position k in the queue. The set of all such bijections is denoted by $\Pi(M)$. We assume that the machine starts processing at time 0 and that there is an initial order on the jobs, $\sigma_0 \in \Pi(M)$, before the processing of the machine starts. We denote by p_j the processing time of job j , i.e., the time that the machine needs to execute job j . We assume $p_j \geq 0$ for every $j \in M$. Furthermore, we assume that, given a processing order, the jobs are processed in a semi-active way. Here, a processing order is called semi-active if there does

not exist a job that could be processed earlier without altering the processing order.

If the jobs are processed according to the order σ in a semi-active way, then the completion time of job j is given by

$$C_j^\sigma = \sum_{k \in M: \sigma(k) \leq \sigma(j)} p_k.$$

The costs of player i are assumed to be described by a cost function which depends only (in a weakly monotonic way) on the completion times of the jobs in $J(i)$, i.e., $c_i : [0, \infty)^{J(i)} \rightarrow \mathbb{R}$. Given $\sigma \in \Pi(M)$, we denote $c_i(\sigma) = c_i((C_j^\sigma)_{j \in J(i)})$.

A *sequencing situation with repeated players* (an RP-sequencing situation) is a 6-tuple $(N, M, J, \sigma_0, p, c)$ with $p = (p_j)_{j \in M}$ and $c = (c_i)_{i \in N}$. Costs are assumed to be additive across players. Therefore, given a processing order σ , the total costs of coalition $S \subseteq N$, when its members decide to cooperate, equals the sum of individual costs of the members of S :

$$c_S(\sigma) = \sum_{i \in S} c_i(\sigma).$$

In order to determine the minimal costs of coalition S , we need to fix which processing orders of jobs are admissible for coalition S with respect to the initial order σ_0 . Let $J_c(S) = \{j \in M : N(j) \subseteq S\}$ be the set of jobs in which only (some) members of S are involved, i.e., $J_c(S)$ is the set of jobs that are controlled by coalition S . An order $\sigma \in \Pi(M)$ is *admissible* for S if

$$P_j(\sigma) = P_j(\sigma_0) \text{ for all } j \notin J_c(S),$$

where $P_j(\sigma) = \{k \in M : \sigma(k) < \sigma(j)\}$ is the set of predecessors of job j with respect to σ . We denote by $\mathcal{A}(S)$ the set of admissible orders for coalition S . In particular, note that if an order is admissible for S , the completion time of each job in which a player of $N \setminus S$ is involved does not change. Moreover, only within connected parts of $J_c(S)$ w.r.t. σ_0 , jobs can be reordered.

Summarizing, a coalition S faces the following optimization problem:

$$\min_{\sigma \in \mathcal{A}(S)} c_S(\sigma).$$

Clearly, there exists an order for which the joint costs of players in S are minimized since $\mathcal{A}(S)$ is finite. Note that $\mathcal{A}(N) = \Pi(M)$.

Given an RP-sequencing situation, $(N, M, J, \sigma_0, p, c)$, we define the associated *RP-sequencing (cost savings) game* (N, v) by

$$v(S) = c_S(\sigma_0) - \min_{\sigma \in \mathcal{A}(S)} c_S(\sigma) = \max_{\sigma \in \mathcal{A}(S)} (c_S(\sigma_0) - c_S(\sigma)), \text{ for every } S \subseteq N,$$

i.e., the value of coalition S equals the maximal cost savings that the coalition can obtain by means of admissible rearrangements.

Example 2.1. Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation with $N = \{1, 2, 3\}$, $M = \{A, B, C, D\}$ and $J(1) = \{A, B\}$, $J(2) = \{B, C\}$, $J(3) = \{A, D\}$. Let the initial order of the jobs, σ_0 , be A-B-C-D and let the processing times vector be $p = (1, 2, 1, 1)$. This situation is depicted below:

A	B	C	D	
1,3	1,2	2	3	
0	1	3	4	5

The cost functions of the players are defined as follows:

$$c_1(\sigma) = C_A^\sigma + C_B^\sigma, \quad c_2(\sigma) = 2 \min\{C_B^\sigma, C_C^\sigma\}, \quad \text{and} \quad c_3(\sigma) = \max\{C_A^\sigma, C_D^\sigma\}.$$

In this case player 1 pays the sum of the processing time of his his two jobs. For player 2 his jobs are substitutes and his cost function depends linearly on the completion time of the job which is processed first. For player 3 the jobs are complementary and his costs depend linearly on the completion time of the job which is processed last. The values of the associated RP-sequencing game are in Table 1.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
$v(S)$	0	0	0	1	0	1	3

Table 1: values of the RP-sequencing game.

As an illustration we compute the value of coalition $\{1, 2\}$. For this purpose we need the set of admissible orders for coalition $\{1, 2\}$. Since $J_c(S) = \{B, C\}$, $\mathcal{A}(S) = \{\sigma_0, \sigma\}$, where σ corresponds to A-C-B-D. Then,

$$c_1(\sigma_0) = 1 + 3 = 4, \quad c_2(\sigma_0) = 2 \cdot 3 = 6$$

and

$$c_1(\sigma) = 1 + 4 = 5, \quad c_2(\sigma) = 2 \cdot 2 = 4.$$

Hence,

$$v(\{1, 2\}) = \max_{\sigma \in \mathcal{A}(S)} (c_S(\sigma_0) - c_S(\sigma)) = \max(0, (4 + 6) - (5 + 4)) = 1.$$

Note that $J_c(\{1, 3\}) = \{A, D\}$, which is unconnected w.r.t. σ_0 , so $\mathcal{A}(\{1, 3\}) = \{\sigma_0\}$. One readily checks that the optimal order for coalition $\{2, 3\}$ is A-B-D-C and for the grand coalition it is C-A-D-B. \square

3 Balancedness of RP-sequencing games

In this section we show that RP-sequencing games are balanced if the underlying cost functions satisfy additivity with respect to the initial order. In fact, introducing so called job games we will prove that these games are balanced and that each core element of a job game gives rise to a core element of the corresponding RP-sequencing game.

Recall that the *core* of a cooperative game (N, v) is given by

$$\text{Core}(v) = \{x \in \mathbb{R}^N \mid \sum_{i \in N} x_i = v(N), \sum_{i \in S} x_i \geq v(S) \text{ for all } S \in 2^N\}.$$

A *game* is said to be *balanced* (see Bondareva (1963) and Shapley (1967)) if the core is nonempty.

First we provide an example of an RP-sequencing situation with an empty core.

Example 3.1. Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation with $N = \{1, 2, 3\}$, $M = \{A, B, C, D\}$ and $J(1) = \{A, D\}$, $J(2) = \{B\}$, $J(3) = \{C\}$. Let the initial order σ_0 be A-B-C-D and let the processing times vector be $p = (1, 1, 1, 1)$.

A	B	C	D
1	2	3	1
0	1	2	3
			4

The cost functions of the players are defined as follows:

$$c_1(\sigma) = \begin{cases} 2 & \text{if } C_A^\sigma + C_D^\sigma \leq 4, \\ 8 & \text{if } 4 < C_A^\sigma + C_D^\sigma \leq 5, \\ 10 & \text{if } 5 < C_A^\sigma + C_D^\sigma \leq 6, \\ 20 & \text{if } C_A^\sigma + C_D^\sigma > 6. \end{cases} \quad c_2(\sigma) = \begin{cases} 0 & \text{if } C_B^\sigma \leq 1, \\ 10 & \text{if } 1 < C_B^\sigma \leq 3, \\ 20 & \text{if } C_B^\sigma > 3. \end{cases} \quad c_3(\sigma) = \begin{cases} 0 & \text{if } C_C^\sigma \leq 2, \\ 10 & \text{if } C_C^\sigma > 2. \end{cases}$$

The values of the associated RP-sequencing game are given in Table 2.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
$v(S)$	0	0	0	8	6	10	10

Table 2: values of the RP-sequencing game.

Clearly, $\text{Core}(v) = \emptyset$. □

We now introduce the notion of an RP-job game.

Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation. A rearrangement $\sigma \in \Pi(M)$ of jobs is called *feasible* for a subset $K \subseteq M$ of jobs if

$$P_j(\sigma) = P_j(\sigma_0) \text{ for all } j \in M \setminus K.$$

We denote by $\mathcal{F}(K)$ the set of feasible rearrangements for $K \subseteq M$.

Given an RP-sequencing situation, $(N, M, J, \sigma_0, p, c)$, we define the associated RP-job (cost savings) game, (M, v_J) , by

$$v_J(K) = \max_{\sigma \in \mathcal{F}(K)} \sum_{i \in N(K)} (c_i(\sigma_0) - c_i(\sigma)) \quad \text{for every } K \subseteq M,$$

where $N(K) = \bigcup_{j \in K} N(j)$ is the set of players that are involved in K . Note that $N(\{j\}) = N(j)$. Further, note that $v_J(M) = v(N)$ and $v_J(J_c(S)) = v(S)$ for every $S \subseteq N$ since $\mathcal{F}(J_c(S)) = \mathcal{A}(S)$, $N(J_c(S)) \subseteq S$, and if $k \in S \setminus N(J_c(S))$, then $c_k(\sigma_0) - c_k(\sigma) = 0$ for every $\sigma \in \mathcal{F}(J_c(S))$.

For $K \subseteq M$, $\hat{\sigma} \in \mathcal{F}(K)$ is called an *optimal order for K* if it maximizes the expression $\sum_{i \in N(K)} (c_i(\sigma_0) - c_i(\sigma))$ over $\sigma \in \mathcal{F}(K)$.

Example 3.2. The values of the RP-job game associated to the RP-sequencing situation in Example 2.1 are: $v_J(\{j\}) = 0$ for every $j \in M$, $v_J(\{A, B\}) = v_J(\{B, C\}) = v_J(\{C, D\}) = 1$, $v_J(\{A, C\}) = v_J(\{A, D\}) = v_J(\{B, D\}) = 0$, $v_J(\{A, B, C\}) = v_J(\{B, C, D\}) = 2$, $v_J(\{A, B, D\}) = v_J(\{A, C, D\}) = 1$, $v_J(\{A, B, C, D\}) = 3$.

One readily checks that, $(1, 1, 1, 0) \in \text{Core}(v_J)$. \square

Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation and let $i \in N$. A cost function c_i is *additive with respect to σ_0* if for all $L_1, L_2 \subseteq M$ with $L_1 \cap L_2 = \emptyset$, all $\rho \in \mathcal{F}(L_1)$ and $\tau \in \mathcal{F}(L_2)$, it holds that

$$c_i(\sigma_0) - c_i(\pi) = (c_i(\sigma_0) - c_i(\rho)) + (c_i(\sigma_0) - c_i(\tau)),$$

where $\pi \in \mathcal{F}(L_1 \cup L_2)$ is such that $\pi(j) = \begin{cases} \rho(j) & \text{if } j \in L_1, \\ \tau(j) & \text{if } j \in L_2, \\ \sigma_0(j) & \text{if } j \in M \setminus (L_1 \cup L_2). \end{cases}$

In Curiel et al. (1994) sequencing games with weakly increasing cost functions are studied, which satisfy additivity with respect to the initial order since the set of jobs is identified with the set of players and the correspondence J is bijective. Cost functions c_i which are linear with respect to the sum, the minimum or the maximum of the completion times in $J(i)$, as those in Example 2.1, are examples of additive cost functions with respect to the initial order. In Example 3.1 the cost functions of player 1 is not additive with respect to σ_0 . Another type of cost functions that are additive with respect to the initial order are cost functions c_i of the form $c_i(\sigma) = \sum_{j \in J(i)} \kappa_j(\sigma)$ where $\kappa_j(\sigma)$ is a weakly monotonic cost function on the completion time of job j .

Next, we consider Example 3.1 to illustrate that if cost functions are not additive with respect to the initial order the corresponding RP-job game needs not be balanced.

Example 3.3. Consider the RP-sequencing situation introduced in Example 3.1. It is readily to check that for $L_1 = \{A, B\}$, $L_2 = \{C, D\}$, $\rho : \text{B-A-C-D}$, $\tau : \text{A-B-D-C}$, $\pi : \text{B-A-D-C}$, it holds:

$$c_1(\sigma_0) - c_1(\pi) = 8 - 8 = 0 < 4 = (8 - 10) + (8 - 2) = (c_1(\sigma_0) - c_1(\rho)) + (c_1(\sigma_0) - c_1(\tau)).$$

Moreover, the associated RP-job game is: $V_J(\{A\}) = V_J(\{B\}) = V_J(\{C\}) = V_J(\{D\}) = 0$, $V_J(\{A, B\}) = 8$, $V_J(\{A, C\}) = V_J(\{A, D\}) = 0$, $V_J(\{B, C\}) = 10$, $V_J(\{B, D\}) = 0$, $V_J(\{C, D\}) = 6$, $V_J(\{A, B, C\}) = V_J(\{B, C, D\}) = 10$, $V_J(\{A, B, D\}) = 8$, $V_J(\{A, C, D\}) = 6$, $V_J(M) = 10$.

In this case, $\text{Core}(v_J) = \emptyset$ because $v_J(\{A, B\}) + v_J(\{C, D\}) = 8 + 6 > 10 = v_J(N)$.

In the following lemma we show that a job game is σ_0 -component additive if all underlying cost functions are additive with respect to σ_0 .

Curiel et al. (1994) introduced the class of σ_0 -component additive games and proved that they are balanced. Given an order $\sigma_0 \in \Pi(N)$, a game (N, v) is called σ_0 -component additive if the following three conditions are satisfied,

- (1) $v(\{i\}) = 0$ for all $i \in N$,
- (2) (N, v) is super-additive (i.e., $v(S) + v(T) \leq v(S \cup T)$ for all $S, T \subseteq N$ such that $S \cap T = \emptyset$),
- (3) $v(S) = \sum_{T \in S/\sigma_0} v(T)$, where S/σ_0 is the set of all maximally connected components of S

according to σ_0 . Here, a *coalition* T is called *connected* with respect to σ_0 if for all $i, j \in T$ and $k \in N$ such that $\sigma_0(i) < \sigma_0(k) < \sigma_0(j)$ it holds that $k \in T$. Notice that S/σ_0 is a partition of S .

Lemma 3.4. Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation in which all cost functions c_i , $i \in N$, are additive with respect to σ_0 . Then the associated RP-job game (M, v_J) is σ_0 -component additive, and hence balanced.

PROOF: By definition of $\mathcal{F}(S)$, $v_J(\{j\}) = 0$ for all $j \in M$. Next, we will show that (N, v_J) is super-additive.

Let $K, L \subseteq M$ be such that $K \cap L = \emptyset$, then

$$\begin{aligned} v_J(K) + v_J(L) &= \sum_{i \in N(K)} (c_i(\sigma_0) - c_i(\sigma_K)) + \sum_{i \in N(L)} (c_i(\sigma_0) - c_i(\sigma_L)) \\ &= \sum_{i \in N(K \cup L)} (c_i(\sigma_0) - c_i(\sigma_{K \cup L})) \\ &\leq \max_{\sigma \in \mathcal{F}(K \cup L)} \sum_{i \in N(K \cup L)} (c_i(\sigma_0) - c_i(\sigma)) \\ &= v_J(K \cup L) \end{aligned}$$

where $\sigma_K \in \mathcal{F}(K)$ is an optimal order for K , $\sigma_L \in \mathcal{F}(L)$ is an optimal order for L , and $\sigma_{K \cup L} \in \mathcal{F}(K \cup L)$ is defined by

$$\sigma_{K \cup L}(j) = \begin{cases} \sigma_K(j) & \text{if } j \in K, \\ \sigma_L(j) & \text{if } j \in L, \\ \sigma_0(j) & \text{if } j \in M \setminus (K \cup L). \end{cases}$$

Note that the second equality holds since the cost functions are additive with respect to σ_0 .

Finally, we need to show that $v_J(K) = \sum_{L \in K/\sigma_0} v_J(L)$. Because K/σ_0 is a partition of K , it is sufficient to show it for K with $K/\sigma_0 = \{L_1, L_2\}$. Hence,

$$\begin{aligned} v(K) &= \max_{\sigma \in \mathcal{F}(K)} \left\{ \sum_{i \in N(K)} (c_i(\sigma_0) - c_i(\sigma)) \right\} \\ &= \max_{\substack{\sigma_1 \in \mathcal{F}(L_1) \\ \sigma_2 \in \mathcal{F}(L_2)}} \left\{ \sum_{i \in N(L_1)} (c_i(\sigma_0) - c_i(\sigma_1)) + \sum_{i \in N(L_2)} (c_i(\sigma_0) - c_i(\sigma_2)) \right\} \\ &= v_J(L_1) + v_J(L_2), \end{aligned}$$

where the second equality holds by definition of $\mathcal{F}(K)$, the fact that L_1 and L_2 are not connected, and by additivity of the cost functions with respect to σ_0 . \square

Now, we can state our main theorem.

Theorem 3.5. Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation in which all cost functions c_i , $i \in N$, are additive with respect to σ_0 . Then the associated RP-sequencing game (N, v) is balanced.

PROOF: By Lemma 3.4, the associated RP-job game, (M, v_J) , is balanced. Let $y \in \mathbb{R}^M$ be a core element of the RP-job game. Define an allocation $x \in \mathbb{R}^N$ by

$$x_i = \sum_{j \in J(i)} \lambda_j^i y_j$$

where, for every $j \in M$ it holds: $\lambda_j \in \mathbb{R}^N$, $\lambda_j^i \geq 0$ for all $i \in N(j)$, $\lambda_j^i = 0$ for all $i \in N \setminus N(j)$ and $\sum_{i \in N} \lambda_j^i = 1$. Note that in this way the cost savings y_j allocated to a particular job $j \in M$ are reallocated only to players in $N(j)$. We will show that $x \in \text{Core}(v)$. First, we show efficiency,

$$\sum_{i \in N} x_i = \sum_{i \in N} \sum_{j \in J(i)} \lambda_j^i y_j = \sum_{j \in M} \sum_{i \in N(j)} \lambda_j^i y_j = \sum_{j \in M} y_j \sum_{i \in N} \lambda_j^i = v_J(M) = v(N)$$

where the second equality follows from $\bigcup_{i \in N} J(i) = M$, the second one holds since $\lambda_j^i = 0$ for all $i \in N \setminus N(j)$, and the fourth equality is satisfied since y is a core element of v_J .

Stability follows from the fact that

$$\begin{aligned} \sum_{i \in S} x_i &= \sum_{i \in S} \sum_{j \in J(i)} \lambda_j^i y_j = \sum_{j \in J(S)} \sum_{i \in N(j) \cap S} \lambda_j^i y_j = \sum_{j \in J(S)} y_j \sum_{i \in N(j) \cap S} \lambda_j^i \geq \sum_{j \in J_c(S)} y_j \sum_{i \in N(j)} \lambda_j^i \\ &= \sum_{j \in J_c(S)} y_j \geq v_J(J_c(S)) = v(S), \end{aligned}$$

for all $S \subseteq N$, where $J(S) := \bigcup_{l \in S} J(l)$. The first inequality holds because $J_c(S) \subseteq J(S)$ and $N(j) \cap S = N(j)$ for all $j \in J_c(S)$ by definition of $J_c(S)$ and the second inequality is due to $y \in \text{Core}(v_J)$. \square

Example 3.6. Using the allocation rule given in the proof of Theorem 3.2, we can find core elements for the RP-sequencing game in Example 2.1.

We will apply the method applied in the proof of Theorem 3.5 to $(1, 1, 1, 0) \in \text{Core}(v_J)$ with $\lambda_A = (\frac{1}{2}, 0, \frac{1}{2})$, $\lambda_B = (\frac{1}{2}, \frac{1}{2}, 0)$, $\lambda_C = (0, 1, 0)$, $\lambda_D = (0, 0, 1)$, in this case $x = (\frac{1}{2} + \frac{1}{2}, \frac{1}{2} + 1, \frac{1}{2} + 0) = (1, \frac{3}{2}, \frac{1}{2}) \in \text{Core}(v)$. \square

A special, well investigated subclass of balanced games is the class of convex games. Convex games were introduced by Shapley (1971). A game (N, v) is said to be *convex* if $v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S)$ for every $S \subseteq T \subseteq N \setminus \{i\}$.

The following example shows a non-convex RP-sequencing game where the cost functions satisfy additivity with respect to the initial order, which illustrates that this property is not sufficient for convexity.

Example 3.7. Let $(N, M, J, \sigma_0, p, c)$ be an RP-sequencing situation with $N = \{1, 2, 3, 4, 5\}$, $M = \{A, B, C, D, E, F, G\}$ and $J(1) = \{D, G\}$, $J(2) = \{A, E\}$, $J(3) = \{C\}$, $J(4) = \{B\}$, $J(5) = \{F\}$. Let the initial order of the jobs, σ_0 , be A-B-C-D-E-F-G and let the processing times vector be $p = (3, 1, 6, 1, 1, 1, 1)$. This situation is depicted below:

A	B	C	D	E	F	G
2	4	3	1	2	5	1
0	3	4	10	11	12	13
						14

The cost functions of the players are defined as follows:

$$c_1(\sigma) = \max_{j \in J(1)} \{C_j^\sigma\}, c_2(\sigma) = 4 \max_{j \in J(2)} \{C_j^\sigma\}, c_3(\sigma) = 6 \min_{j \in J(3)} \{C_j^\sigma\}, c_4(\sigma) = 10C_B^\sigma, \text{ and } c_5(\sigma) = C_F^\sigma.$$

Let $S = \{2, 3\}$, $T = \{2, 3, 4\}$, and $i = 1$. It is readily checked that the optimal order for coalition S is σ_0 and therefore $v(S) = 0$, the optimal order for coalition $S \cup \{i\}$ is A-B-E-C-D-F-G and $v(S \cup \{i\}) = 22$, the optimal order for coalition T is B-C-A-D-E-F-G and $v(T) = 48$, finally, the optimal order for coalition $T \cup \{i\}$ is B-C-A-E-D-F-G and $v(T \cup \{i\}) = 52$. Therefore, $v(T \cup \{i\}) - v(T) = 4 < 22 = v(S \cup \{i\}) - v(S)$ and the game is not convex. \square

References

- Bondareva, O. N. (1963), ‘Some applications of linear programming methods to the theory of cooperative games’, *Problemy Kibernetiki* **10**, 119–139 (in Russian).
- Borm, P., Fiestras-Janeiro, G., Hamers, H., Sánchez, E. and Voorneveld, M. (2002), ‘On the convexity of games corresponding to sequencing situations with due dates’, *European Journal of Operational Research* **136**, 616–634.
- Calleja, P., Borm, P., Hamers, H., Klijn, F. and Slikker, M. (2002), ‘On a class of parallel sequencing situations and related games’, *Annals of Operations Research* **109**, 263–276.
- Curiel, I., Hamers, H. and Klijn, F. (2002), Sequencing games: a survey, in P. Borm and H. Peters, eds, ‘Chapters in Game Theory: in Honor of Stef Tijs’, Kluwer Academic Publishers, Boston, pp. 27–50.
- Curiel, I., Pederzoli, G. and Tijs, S. (1989), ‘Sequencing games’, *European Journal of Operational Research* **40**, 344–351.
- Curiel, I., Potters, J., Prasad, R., Tijs, S. and Veltman, B. (1994), ‘Sequencing and cooperation’, *Operations Research* **42**, 566–568.
- Hamers, H., Borm, P. and Tijs, S. (1995), ‘On games corresponding to sequencing situations with ready times’, *Mathematical Programming* **69**, 471–483.
- Hamers, H., Klijn, F. and Suijs, J. (1999), ‘On the balancedness of m-sequencing games’, *European Journal of Operational Research* **119**, 678–691.

- Shapley, L. S. (1967), ‘On balanced sets and cores’, *Naval Research Logistics Quarterly* **14**, 453–460.
- Shapley, L. S. (1971), ‘Cores of convex games’, *International Journal of Game Theory* **1**, 11–26.
- Slikker, M. (2003), ‘Relaxed sequencing games have a nonempty core’, *Beta working paper WP-101*. Technische Universiteit Eindhoven, Eindhoven, The Netherlands.
- van den Nouweland, A., Krabbenborg, M. and Potters, J. (1990), ‘Flow-shops with a dominant machine’, *European Journal of Operational Research* **62**, 38–46.
- van Velzen, B. and Hamers, H. (2003), ‘On the balancedness of relaxed sequencing games’, *Mathematical Methods of Operations Research* **57**(2), 287–297.